

INestMicroservice

August 18, 2026

INestMicroservice

Kind: Interface

Source: [packages/common/interfaces/nest-microservice.interface.ts](#)

Part of: [Common](#)

Interface describing Microservice Context.

`INestMicroservice` describes the runtime context of a NestJS microservice application. It exposes lifecycle state through an RxJS observable, allowing integrations to react to status changes such as startup, readiness, or shutdown.

Properties

Property	Type
<code>status</code>	<code>Observable<string></code>

Diagram

```
graph LR
  A[Microservice Bootstrap] --> B[INestMicroservice]
  B --> C[status: Observable<string>]
  C --> D[Lifecycle State Subscribers]
  D --> E[Monitoring / Shutdown / Integration Logic]
```

Usage

```
import { INestMicroservice } from '@nestjs/common';

async function monitorMicroservice(app: INestMicroservice) {
  app.status.subscribe((status) => {
    console.log(`Microservice status: ${status}`);
  });
}
```

```
await app.listen();  
}
```

AI Coding Instructions

- Treat `status` as an RxJS `Observable` ; subscribe to it rather than reading it as a synchronous string value.
- Clean up long-lived subscriptions when adding custom lifecycle monitoring to avoid memory leaks.
- Use the interface when accepting a microservice application context in reusable helpers or integration code.
- Keep lifecycle-dependent logic resilient to multiple status emissions, including startup and shutdown transitions.

Used by

4 references from 4 files. Each is a place in this repository where the symbol is actually used — go read one rather than trusting an example.

Imported by (4)

- `NestApplication` — `packages/core/nest-application.ts` :54
- `IEntryNestModule` — `packages/core/nest-factory.ts` :39
- `NestMicroservice` — `packages/microservices/nest-microservice.ts` :35
- `TestingModule` — `packages/testing/testing-module.ts` :26